

ITA\_利用手順マニュアル

RestAPI

*－*第1.8版*－*

Copyright © NEC Corporation 2019. All rights reserved.

免責事項

本書の内容はすべて日本電気株式会社が所有する著作権に保護されています。

本書の内容の一部または全部を無断で転載および複写することは禁止されています。

本書の内容は将来予告なしに変更することがあります。

日本電気株式会社は、本書の技術的もしくは編集上の間違い、欠落について、一切責任を負いません。

日本電気株式会社は、本書の内容に関し、その正確性、有用性、確実性その他いかなる保証もいたしません。

商標

* LinuxはLinus Torvalds氏の米国およびその他の国における登録商標または商標です。
* Red Hatは、Red Hat, Inc.の米国およびその他の国における登録商標または商標です。
* Apache、Apache Tomcat、Tomcatは、Apache Software Foundationの登録商標または商標です。
* Ansibleは、Red Hat, Inc.の登録商標または商標です。
* Active Directoryは、米国Microsoft Corporationの米国およびその他の国における登録商標または商標です。

その他、本書に記載のシステム名、会社名、製品名は、各社の登録商標もしくは商標です。

なお、® マーク、TMマークは本書に明記しておりません。

※本書では「Exastro IT Automation」を「ITA」として記載します。

# 目次

[目次 2](#_Toc75872236)

[はじめに 4](#_Toc75872237)

[1 ITAシステム　REST APIの概要 5](#_Toc75872238)

[1.1 REST APIについて 6](#_Toc75872239)

[2 標準REST機能の利用 7](#_Toc75872240)

[2.1 リクエストの形式 8](#_Toc75872241)

[2.2 利用可能なメソッドとコマンド 10](#_Toc75872242)

[(1) GET（Method） 10](#_Toc75872243)

[(2) INFO(X-Command) 12](#_Toc75872244)

[(3) FILTER・FILTER\_DATAONLY (X-Command) 12](#_Toc75872245)

[(4) EDIT(X-Command) 16](#_Toc75872246)

[3 メニューエクスポート/インポート利用編 23](#_Toc75872247)

[3.1 メニューエクスポートを対象としたRestAPI 24](#_Toc75872248)

[3.1.1 リクエストの形式 24](#_Toc75872249)

[3.1.2 INFO 24](#_Toc75872250)

[3.1.3 EXECUTE 26](#_Toc75872251)

[3.2 メニューインポートを対象としたRestAPI 28](#_Toc75872252)

[3.2.1 リクエストの形式 28](#_Toc75872253)

[3.2.2 UPLOAD 30](#_Toc75872254)

[3.2.3 EXECUTE 31](#_Toc75872255)

[4 Symphony利用編 33](#_Toc75872256)

[4.1 Symphony登録作業を対象としたRestAPI 34](#_Toc75872257)

[4.1.1 リクエストの形式 34](#_Toc75872258)

[4.1.2 INFO 34](#_Toc75872259)

[4.1.3 FILTER・FILTER\_DATAONLY 35](#_Toc75872260)

[4.1.4 EDIT 35](#_Toc75872261)

[4.2 Symphony作業実行を対象としたRestAPI 39](#_Toc75872262)

[4.2.1 リクエストの形式 39](#_Toc75872263)

[4.2.2 レスポンスの項目 39](#_Toc75872264)

[4.2.3 EXECUTE 40](#_Toc75872265)

[4.2.4 CANCEL 41](#_Toc75872266)

[4.2.5 SCRAM 42](#_Toc75872267)

[4.2.6 RELEASE 42](#_Toc75872268)

[4.3 Symphony作業確認を対象としたRestAPI 44](#_Toc75872269)

[4.3.1 リクエストの形式 44](#_Toc75872270)

[4.3.2 レスポンスの項目 44](#_Toc75872271)

[4.3.3 INFO 45](#_Toc75872272)

[5 Conductor利用編 48](#_Toc75872273)

[5.1 Conductor登録作業を対象としたRestAPI 49](#_Toc75872274)

[5.1.1 リクエストの形式 49](#_Toc75872275)

[5.1.2 INFO 50](#_Toc75872276)

[5.1.3 FILTER・FILTER\_DATAONLY 53](#_Toc75872277)

[5.1.4 EDIT 57](#_Toc75872278)

[5.2 Conductor作業実行を対象としたRestAPI 63](#_Toc75872279)

[5.2.1 リクエストの形式 63](#_Toc75872280)

[5.2.2 レスポンスの項目 63](#_Toc75872281)

[5.2.3 EXECUTE 64](#_Toc75872282)

[5.2.4 CANCEL 65](#_Toc75872283)

[5.2.5 SCRAM 66](#_Toc75872284)

[5.2.6 RELEASE 66](#_Toc75872285)

[5.3 Conductor作業確認を対象としたRestAPI 68](#_Toc75872286)

[5.3.1 リクエストの形式 68](#_Toc75872287)

[5.3.2 レスポンスの項目 68](#_Toc75872288)

[5.3.3 INFO 69](#_Toc75872289)

[6 Movement利用編 72](#_Toc75872290)

[6.1 Movement作業実行を対象としたRestAPI 73](#_Toc75872291)

[6.1.1 リクエストの形式 73](#_Toc75872292)

[6.1.2 レスポンスの項目 74](#_Toc75872293)

[6.1.3 EXECUTE 74](#_Toc75872294)

[6.1.4 CANCEL 75](#_Toc75872295)

[6.1.5 SCRAM 75](#_Toc75872296)

[7 付録 77](#_Toc75872297)

[7.1 トラブルシューティング 78](#_Toc75872298)

# はじめに

本書は、ITAシステムのRestAPIの概要および操作方法について説明します。

# ITAシステム　REST APIの概要

本章ではITAを操作するための標準REST APIについて説明します。

## REST APIについて

ITAでは、外部プログラムからITAで管理されているリソースに対して、各種操作を行うことが可能なREST APIを提供しています。

* 以下の表に記載しているメニュー以外はすべて標準的なRESTAPIを使用できます。

標準的なRESTAPIについて、「2標準REST機能の利用」を参照

表 1‑1個別のREST API一覧

|  |  |  |  |
| --- | --- | --- | --- |
| メニューグループ | メニュー名 | メニューID | 参照先 |
| Symphony | Symphonyクラス編集 | 2100000306 | 4 Symphony利用編 |
| Symphony作業実行 | 2100000308 |
| Symphony作業確認 | 2100000309 |
| Conductor | Conductorクラス編集 | 2100180003 | 5 Conductor利用編 |
| Conductor作業実行 | 2100180004 |
| Conductor作業確認 | 2100180005 |
| エクスポート/インポート | メニューエクスポート | 2100000211 | 3 メニューエクスポート/インポート利用編 |
| メニューインポート | 2100000212 |
| Ansible-Legacy | 作業実行 | 2100020111 | 6　Movement利用編 |
| 作業状態確認 | 2100020112 |
| Ansible-Pioneer | 作業実行 | 2100020211 |
| 作業状態確認 | 2100020212 |
| Ansible-LegacyRole | 作業実行 | 2100020312 |
| 作業状態確認 | 2100020313 |
| Terraform | 作業実行 | 2100080009 |
| 作業状態確認 | 2100080010 |

# 標準REST機能の利用

外部プログラムからREST APIを利用して、ITAで管理されているリソースに対し、操作を行うことが可能です。以下に、呼び出し規約を示します。

## リクエストの形式

ITAのREST APIでは、ITA上の各メニューのパスに対してHTTPリクエストを発行します。

パス：

https://<HostName>:<Port>/default/menu/07\_rest\_api\_ver1.php?no=（各メニューのメニューID）

例）[管理コンソール]-[システム設定]メニュー（メニューID：2100000202）の場合

https:// exastro-it-automation:443/default/menu/07\_rest\_api\_ver1.php?no=2100000202

※<HostName>:ITAインストーラでインストールした時のホスト名「exastro-it-automation」

HTTPヘッダ：

以下の表にあるものが利用可能です。

表 2‑1 HTTPヘッダパラメータ一覧

| HTTPヘッダ | 説明 |
| --- | --- |
| Host | ITAのRestAPIサーバーのホスト名またはIPアドレスとポート番号をコロン(:)区切りで指定する。 |
| Content-Type | “application/json”を指定する。  MethodがGETの場合は指定しなくてもよい。 |
| Authorization | ITAの認証要メニューにアクセスする場合は、  「ログインID」と「パスワード」\* を、半角コロン(:)で結合して、base64encodeをした値、を指定。  MethodがGETの場合は指定しなくてもよい。 |
| X-Command | MethodがPOSTの場合のみに設定可能。  【INFO】、【FILTER】、【FILTER\_DATAONLY】、【EDIT】 のいずれかを設定できる。 |

HTTPヘッダの文字列は大文字でも小文字でも問題ありません。

\* ITAのパスワードが期限切れとなっていた場合、RestAPIはErrorとなります。

Webシステムのログイン画面から、パスワードを変更してからリクエストを行ってください。

但し、ActiveDirectory連携機能を利用している場合は、ActiveDirectory上で管理される認証情報に

従います。（※ActiveDirectory連携機能の連携対象外ユーザーは、この限りではありません）

**ActiveDirectory連携機能の詳細は、「利用手順マニュアル\_管理コンソール」-「ActiveDirectory連携機能の利用」をご参照ください。**

HTTPヘッダの例：

ログインIDが[test\_loginid]で、パスワードが[test\_password]の場合

test\_loginid: test\_password　を、base64encodeで暗号化

→[qTImqS9fo2qcozyxBaEyp3EspTSmp3qipzD=]）

Host:<HostName>:<Port>

Content-Type:application/json

Authorization: qTImqS9fo2qcozyxBaEyp3EspTSmp3qipzD=

X-Command: INFO

## 利用可能なメソッドとコマンド

HTTPリクエストメソッドは、原則POSTを指定してください。

例外として、ITAで認証不要と設定したメニューにアクセスする場合のみGETを指定できます。

利用可能なメソッドとコマンドの階層は以下のとおりです。

Method

　　　　　├GET ‐　①

　　　　　｜

　　　　　└POST

(X-Command)｜

├【INFO】 ‐　②

　　　　　　　├【FILTER】 ‐　③

　　　　　　　├【FILTER\_DATAONLY】 ‐　④

　　　　　　　└【EDIT】 ‐　⑤

#### GET（Method）

列情報（列番号と列名）および、通常ステータス（廃止または活性中）の全レコードの行数とレコード内容を返却します。

・HTTPヘッダ

なし

・contentパラメータ

なし

・レスポンス

1)レコード行数

（JSON形式）

キー{resultdata} -> キー{CONTENTS} -> キー{RECORD\_LENGTH}の中に、

数値として格納されます。

2)列情報（列番号と列名）

（JSON形式）

キー{resultdata} -> キー{CONTENTS} -> キー{BODY} -> キー{0}の中に、

0から始まる数値を、キーとする配列として格納されます。

表 2‑2レスポンスパラメータ一覧(列情報)

|  |  |
| --- | --- |
| 列番号 | 列名 |
| 0 | 一列目 |
| 1 | 二列目 |
| … | … |

3)レコード情報

（JSON形式）（1行につき1個の配列（列番号と列別データ））

キー{resultdata} -> キー{CONTENTS} -> キー{BODY} -> キー{(1以降、

該当レコードの存在行数を上限とする数値)}の中に、0から始まる数値をキーとする配列として格納されます。

表 2‑3レスポンスパラメータ一覧(レコード情報)

|  |  |
| --- | --- |
| 列番号 | 列データ |
| 0 | 一列目データ配列 |
| 1 | 二列目データ配列 |
| … | … |

Method：Getでレスポンスされるデータの表と、Jsonの階層構造を以下に示します。

表 2‑4 返されるデータ一覧

|  |  |  |  |
| --- | --- | --- | --- |
|  | 0 | 1 | 2 |
| 0 | A | I | U |
| 1 | あ | い | う |
| 2 | か | き | く |
| 3 | さ | し | す |

|  |
| --- |
| ▽JSON形式  {  "resultdata": {  "CONTENTS": {  "├RECORD\_LENGTH": 3,  "└BODY": {  "0": [  "A",  "I",  "U"  ],  "1": [  "あ",  "い",  "う"  ],  "2": [  "か",  "き",  "く"  ],  "3": [  "さ",  "し",  "す"  ]  }  }  }  } |

#### INFO(X-Command)

列情報のみを取得します。

X-Command(FILTER、FILTER\_DATAONLY)またはX-Command(EDIT)を実行する際に必要な情報を取得することができます。

・HTTPヘッダ

表 2‑5 HTTPヘッダパラメータ一覧

|  |  |
| --- | --- |
| HTTPヘッダ | 値 |
| X-Command | INFO |

・contentパラメータ

なし

・レスポンス

1)列情報（列番号と列名）

（JSON形式）

キー{resultdata} -> キー{CONTENTS} -> キー{INFO}の中に、

0から始まる数値をキーとする配列として格納されます。

resultdata

└CONTENTS

└INFO

├0: 一列目列名

├1:　二列目列名

└2:　三列目列名

#### FILTER・FILTER\_DATAONLY (X-Command)

パラメータで指定した条件に合致したレコードの、列情報（列番号と列名）および、

通常ステータス（廃止または活性中）の全レコードの行数とレコード内容を返却します。

FILTER\_DATAONLYを指定した場合は、列情報を省略します。

表 2‑6 HTTPヘッダパラメータ一覧

|  |  |
| --- | --- |
| HTTPヘッダ | 値 |
| X-Command | FILTER |
| FILTER\_DATAONLY |

・contentパラメータ

1）フィルタ形式

操作したいWebページの表示フィルタでフィルタをかけられる列と形式に従い、

列ごとに以下をフィルタの種類として指定可能です。

・NORMAL ‐ 通常のLIKE検索

・RANGE ‐ 1~5等の範囲検索

また、Webページの表示フィルタにプルダウンが表示されている列については、LIST（複数の完全一致条件によるOR検索。複数条件を配列に格納し指定）設定可能です。

2)指定形式

JSON形式で指定。フィルタ種類ごとの形式で、フィルタ条件を格納します。

JSON形式にする前段階では、１つの連想配列の中に、列番号ごとの連想配列を入れ子にする形で指定します。

複数の列番号ごとの連想配列を格納した場合、ANDで繋いだ意味になります。

さらに列番号ごとの連想配列の中にフィルタ条件の形式と条件をセットにした連想配列を格納してください。列番号ごとの連想配列に複数のフィルタ条件の連想配列を格納した場合、ORでつないだ意味になります。

・パラメータの記述例

|  |
| --- |
| 例）　 FILTERパラメータの記述 |
| 列番号2が列名【項番】(主キーのカラム列)、列番号4が【備考】のコンテンツの場合で、  　【項番】が5以上かつ、【備考】に「あいう」が入っているレコードを抽出したい場合  ↓抽出イメージ   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | 列番号1 | 列番号2 | 列番号3 | 列番号4 | | 列名 | 【列名1】 | 【項番】 | 【列名3】 | 【備考】 | | 1 | \*\*\*\*\* | 1 | \*\*\*\*\* | あいうえお | | 2 | \*\*\*\*\* | 2 | \*\*\*\*\* | かきくけこ | | 3 | \*\*\*\*\* | 3 | \*\*\*\*\* | あいうえお | | 4 | \*\*\*\*\* | 4 | \*\*\*\*\* | かきくけこ | | 5 | \*\*\*\*\* | 5 | \*\*\*\*\* | かきくけこ | | 6 | \*\*\*\*\* | 6 | \*\*\*\*\* | あいうえお | | … | … | … | … | … |   ▽JSON形式  {  "2": {  "RANGE": {  "START": 5  }  },  "4": {  "'NORMAL": "あいう"  }  } |

|  |
| --- |
| 例）　 FILTERパラメータの記述(2) |
| 列番号2が列名【項番】(主キーのカラム列)、のコンテンツの場合で、  　【項番】が10から99の範囲、または、【項番】が1または2または5、のレコードを抽出したい場合。  ↓抽出イメージ   |  |  |  |  | | --- | --- | --- | --- | |  | 列番号1 | 列番2 | … | | 列名 | 【列名１】 | 【項番】 | … | | 1 | \*\*\*\*\* | 1 | … | | 2 | \*\*\*\*\* | 2 | … | | 3 | \*\*\*\*\* | 3 | … | | 4 | \*\*\*\*\* | 4 | … | | 5 | \*\*\*\*\* | 5 | … | | … | … | … | … | | 10 | \*\*\*\*\* | 10 | … | | … | … | … | … | | 99 | \*\*\*\*\* | 99 | … | | … | … | … | … |   ▽JSON形式  {  "2": {  "RANGE": {  "START": "10",  "END": "99"  },  "LIST": [  "1",  "2",  "5"  ]  }  } |

|  |
| --- |
| 例）　 FILTERパラメータの記述(3) |
| 列番号2が列名【項番】(主キーのカラム列)、列番号5が列名【最終更新日付】(日付型/日時型)のコンテンツの場合で、【項番】が1から100で、かつ、【最終更新日付】が2016年8月01日(00:00:00)から2016年12月31日(23:59:59)の範囲のレコードを抽出したい場合  ▽JSON形式  {  "2": {  "RANGE": {  "START": "1",  "END": "100"  }  },  "5": {  "RANGE": {  "START": "2016/08/01 00:00:00",  "END": "2016/12/31 23:59:59"  }  }  } |

・レスポンス

1)レコード行数

（JSON形式）

キー{resultdata} -> キー{CONTENTS} -> キー{RECORD\_LENGTH}の中に、

数値として格納されます。

2)列情報（列番号と列名）※FILTER指定時のみ

（JSON形式）

キー{resultdata} -> キー{CONTENTS} -> キー{BODY} -> キー{0}の中に、

0から始まる数値を、キーとする配列として格納される。

表 2‑7 レスポンスパラメータ一覧(列情報)

|  |  |
| --- | --- |
| 列番号 | 列名 |
| 0 | 一列目 |
| 1 | 二列目 |
| … | … |

3)レコード情報

（JSON形式）（1行につき1個の配列（列番号と列別データ））

キー{resultdata} -> キー{CONTENTS} -> キー{BODY} -> キー{(1以降、該当レコードの存在行数を上限とする数値)}の中に、0から始まる数値をキーとする配列として格納される。

表 2‑8 レスポンスパラメータ一覧(レコード情報)

|  |  |  |
| --- | --- | --- |
| 列番号 | 列データ | 説明 |
| 0 | 一列目データ |  |
| 1 | 二列目データ |  |
| … | … |  |

※Json格納時の階層構造は、Method:GETと同様です。

アクセス許可ロールの検索方法については「Exastro-ITA\_利用手順マニュアル\_データレコード毎のロールベースアクセス制御」をご参照ください。

#### EDIT(X-Command)

レコードの登録、既存レコードの更新、廃止、復活を行います。

・HTTPヘッダ

表 2‑2 HTTPヘッダパラメータ一覧

|  |  |
| --- | --- |
| HTTPヘッダ | 値 |
| X-Command | EDIT |

・パラメータ

1)指定形式

JSON形式で指定してください。

INFOで取得できる列情報をもとに、１レコードにつき１つの配列で指定し、１レコードを

格納した配列を要素とする配列を、JSON形式でエンコードしたものをHTTPリクエスト

のcontextとして送信してください。

[更新]時は、必須項目であってもパラメータを省略した場合、値を変更せずに更新実行  
します。ただし、データを特定する一意項目(ID、No等)は省略できません。

列番号0の列名【処理種別】には、[登録]、[更新]、[廃止]、[復活]のいずれかを

指定してください。

|  |
| --- |
| **例(1) 　登録** |
| 列番号0が列名【処理種別】、列番1が【廃止】、列番号2が【項番】(主キー役のカラム列)、 　・・・（中略）・・・ 　列番号10が【備考】、列番号11が【最終更新日時】、列番号12が【更新用の最終更新日時】、列番号13が【最終更新者】、のコンテンツの場合に、 　2レコードを追加する場合、 ▽JSON形式  [  {  "0": {  "0": "登録",  "1": "",  "2": "",  ・・・（中略）・・・  "10": "備考",  "11": "",  "12": "",  "13": ""  },  "1": {  "0": "登録",  "1": "",  "2": "",  ・・・（中略）・・・  "10": "備考",  "11": "",  "12": "",  "13": ""  }  }  ] |

|  |
| --- |
| **例(2) 　更新** |
| 列番号0が列名【処理種別】、列番号1が【廃止】、列番号2が【項番】(主キー役のカラム列)、 　・・・（中略）・・・ 　列番号9が【備考】、列番号10が【最終更新日時】、 　列番号11が【更新用の最終更新日時】、列番号12が【最終更新者】、のコンテンツの場合に、 　【項番】10のレコードを更新する場合、 ▽JSON形式 [  {  "0": "更新",  "1": "",  "2": "10",  "9": "備考欄",  "10": "2016/08/01 12:30:45",  "11": "【更新用の最終更新日時】",※  "12": "管理者"  }  ] |

※Method：GET、X-Command：FILTERで取得した、【更新用の最終更新日時】をセット

してください。このデータによって、追い越し更新を防止しています。

【更新用の最終更新日時】は、“T\_“で始まっています。

|  |
| --- |
| **例(3) 　登録（ファイルアップロードあり）** |
| 列番号0が列名【処理種別】、列番1が【廃止】、列番号2が【項番】(主キー役のカラム列)、・・・（中略）・・・ 　列番号5が【備考】、列番号6が【最終更新日時】、列番号7が【更新用の最終更新日時】、列番号8が【最終更新者】、のコンテンツの場合に、  ▽JSON形式  ・　1レコードを追加する場合、ファイルのアップロードあり  {  "0": {  "0": "登録",  "3": "PV05004",  "4": "20191226095004.yml",  "5": "TEST"  }  "UPLOAD\_FILE": [  {  "4": "<対象ファイルをbase64encodeをした値>"  }  ]  }  ・　2レコードを追加する場合、ファイルのアップロードあり {  "0": {  "0": "登録",  "3": "PV05004",  "4": "20191226095004.yml",  "5": "TEST"  },  "1": {  "0": "登録",  "3": "PV15004",  "4": "20191226095004.yml",  "5": "TEST"  },  "UPLOAD\_FILE": [  {  "4": "<対象ファイルをbase64encodeした値>"  },  {  "4": "<対象ファイルをbase64encodeした値>"  }  ]  } |

※UPLOAD\_FILEについて、対象ファイルをbase64encodeした値を指定し、ファイルのアップロードを行います。

※ファイルをアップロードする場合、要素順に「UPLOAD\_FILE」に追加します。

※ファイルアップロード時の最大ファイルサイズの上限を変更する場合は、PHPの設定を変更する必要があります。変更する項目について、「 **7.1トラブルシューティング**」を参照してください。デフォルト値については、別紙「【参考】インストール時の各コンフィグ設定値」を参照してください。

・レスポンス

1)各レコードの処理結果

（JSON形式）

キー{resultdata} -> キー{LIST} -> キー{NORMAL} -> キー{register、update、

delete、error}の中に、配列として格納されます。

表 2‑3 Keyパラメータ一覧

|  |  |  |
| --- | --- | --- |
| **key** | **値の型** |  |
| name | 文字列 | 処理結果種類の名前 |
| ct | 数値 | （処理結果ごとの）レコード件数 |

2)各レコードの処理結果

（JSON形式）

キー{resultdata} -> キー{LIST} -> キー{RAW} -> キー{パラメータとして渡した

レコード番号（列情報を送信しなくてよい）デフォルト設定では、0、から始まる）}の中に、

0から始まる数値を、キーとする配列として格納されます。

表 2‑4 Keyパラメータ一覧

|  |  |  |
| --- | --- | --- |
| **key** | **値の型** |  |
| 0 | 文字列 | 結果コード(別表を参照) |
| 1 | 文字列 | 詳細コード(別表を参照) |
| 2 | 文字列 | エラーメッセージ |

・レスポンス結果階層表

resultdata

└LIST

├NORMAL

｜ ├register: {name:,ct:}

｜ ├update:　{name:,ct:}

｜ ├delete:　 {name:,ct:}

｜ └error: {name:,ct:}

｜

└RAW

├0:{0:,1:,2:}

├1: {0:,1:,2:}

├2: {0:,1:,2:}

├　・

├　・

・

・レスポンス

返されるレスポンスには、JSON形式で格納されています。

|  |
| --- |
| {  "status": "SUCCEED",  "resultdata": {  "LIST": {  "NORMAL": {  "register": {  "name": "登録",  "ct":  },  "update": {  "name": "更新",  "ct":  },  "delete": {  "name": "廃止",  "ct":  },  "revive": {  "name": "復活",  "ct":  },  "error": {  "name": "エラー",  "ct":  }  },  "RAW": [  [  ・・・別表：結果コード／詳細コード　一覧・・・  ],  ・・・（中略）・・・  ]  }  }  } |
|
|

別表：結果コード／詳細コード　一覧

|  |  |  |  |
| --- | --- | --- | --- |
| 処理種別 | 結果コード | 詳細コード | 説明 |
| 登録 | 000 | 201 | 登録の成功。 |
| 登録 | 002 | 000 | 必須項目が未入力。 |
| 登録 | 002 | 000 | レコードと重複している項目がある。 |
| 登録 | 002 | 000 | 重複禁止に違反しているレコードが存在している。 |
| 登録 | 002 | 000 | 入力値の長さが規定のバイト数を超えている。 |
| 登録 | 002 | 000 | 入力値[NULLバイト文字等が含まれた値]が不正。 |
| 登録 | 002 | 000 | 半角整数以外が入力された。 |
| 登録 | 002 | 000 | 値が範囲外。 |
| 登録 | 002 | 000 | 入力された値が最小値を下回っているか最大値を上回っている。 |
| 登録 | 002 | 000 | 入力条件を満たしていない。 |
| 登録 | 002 | 000 | 数値以外が入力された。 |
| 登録 | 002 | 000 | タブと改行が入力された。 |
| 登録 | 002 | 000 | タブが入力された。 |
| 登録 | 002 | 000 | 入力値が範囲外。 |
| 登録 | 002 | 000 | 入力値が、PHP関数(checkdate)で正常に処理できる範囲外。 |
| 登録 | 002 | 000 | 利用できない値が選択された。 |
| 登録 | 002 | 000 | 登録時に指定できない項目(主キー)が指定された。 |
| 登録 | - | - | メンテナンス権限がない。 |
| 更新 | 000 | 200 | 更新の成功。 |
| 更新 | 002 | 000 | 必須項目が未入力。 |
| 更新 | 002 | 000 | レコードと重複している項目がある。 |
| 更新 | 002 | 000 | 重複禁止に違反しているレコードが存在している。 |
| 更新 | 002 | 000 | 入力値の長さが規定のバイト数を超えている。 |
| 更新 | 002 | 000 | 入力値[NULLバイト文字等が含まれた値]が不正。 |
| 更新 | 002 | 000 | 半角整数以外が入力された。 |
| 更新 | 002 | 000 | 値が範囲外。 |
| 更新 | 002 | 000 | 入力された値が最小値を下回っているか最大値を上回っている。 |
| 更新 | 002 | 000 | 入力条件を満たしていない。 |
| 更新 | 002 | 000 | 数値以外が入力された。 |
| 更新 | 002 | 000 | タブと改行が入力された。 |
| 更新 | 002 | 000 | タブが入力された。 |
| 更新 | 002 | 000 | 入力値が範囲外。 |
| 更新 | 002 | 000 | 入力値が、PHP関数(checkdate)で正常に処理できる範囲外。 |
| 更新 | 002 | 000 | 利用できない値が選択された。 |
| 更新 | 003 | 000 | 別セッションからレコードが更新されたため、更新の実行が中止された。 |
| 更新 | 003 | 000 | 廃止済レコードへの更新が実行されようとした。 |
| 更新 | 101 | 000 | 更新対象の行が特定できなかった。 |
| 更新 | - | - | メンテナンス権限がない。 |
| 廃止 | 000 | 210 | 廃止の成功。 |
| 廃止 | 002 | 000 | 入力値の長さが規定のバイト数を超えている。 |
| 廃止 | 002 | 000 | 入力値[NULLバイト文字等が含まれた値]が不正。 |
| 廃止 | 002 | 000 | 入力された値が最小値を下回っているか最大値を上回っている。 |
| 廃止 | 002 | 000 | 入力条件を満たしていない。 |
| 廃止 | 002 | 000 | タブが入力された。 |
| 処理種別 | 結果コード | 詳細コード | 説明 |
| 廃止 | 003 | 000 | 別セッションからレコードが更新されたため、廃止の実行が中止された。 |
| 廃止 | 003 | 000 | 廃止済レコードへの廃止が実行されようとした。 |
| 廃止 | 101 | 000 | 廃止対象の行が特定できなかった。 |
| 廃止 | - | - | メンテナンス権限がない。 |
| 復活 | 000 | 200 | 復活の成功。 |
| 復活 | 002 | 000 | 必須項目が未入力。 |
| 復活 | 002 | 000 | 復活の場合、更新できない項目が更新されようとした。 |
| 復活 | 002 | 000 | レコードと重複している項目がある。 |
| 復活 | 002 | 000 | 重複禁止に違反しているレコードが存在している。 |
| 復活 | 002 | 000 | 入力値の長さが規定のバイト数を超えている。 |
| 復活 | 002 | 000 | 入力値[NULLバイト文字等が含まれた値]が不正。 |
| 復活 | 002 | 000 | 入力された値が最小値を下回っているか最大値を上回っている。 |
| 復活 | 002 | 000 | 入力条件を満たしていない。 |
| 復活 | 002 | 000 | タブが入力された。 |
| 復活 | 003 | 000 | 別セッションからレコードが更新されたため、復活の実行が中止された。 |
| 復活 | 003 | 000 | 復活済レコードへの復活が実行されようとした。 |
| 復活 | 101 | 000 | 復活対象の行が特定できなかった。 |
| 復活 | - | - | メンテナンス権限がない。 |
| 表示 | - | - | バリデーションエラー。 |
| 表示 | - | - | 次のいずれか(全レコード,廃止含まず,廃止のみ)が選択されていない。 |
| - | 000 | 000 | 処理スキップして、次のレコードへ。 |

# メニューエクスポート/インポート利用編

## メニューエクスポートを対象としたRestAPI

メニューエクスポートの操作をRestAPIで行うことができます。

利用可能な機能は、メニューグループ「エクスポート/インポート」の、「メニューエクスポート」メニューに相当する操作です。

表 3‑1 メニュー一覧

|  |  |  |
| --- | --- | --- |
| メニューグループ | メニュー名 | メニューID |
| エクスポート/インポート | メニューエクスポート | 2100000211 |

### リクエストの形式

下記の情報でHTTPリクエストを発行します。

・パス

https://<HostName>:<Port>/default/menu/07\_rest\_api\_ver1.php?no=メニューID

メニューIDは**「表 3‑3　X-Commandに指定可能なパラメータ一覧」**を参照してください**。**

・HTTPヘッダ

表 3‑2 HTTPヘッダパラメータ一覧

|  |  |
| --- | --- |
| HTTPヘッダ | 説明 |
| Content-Type | “application/json”を指定する。 |
| Authorization | ITAの認証要メニューにアクセスする場合は、  「ログインID」と「パスワード」\* を、半角コロン(:)で結合して、base64encodeをした値、を指定。 |
| X-Command | EXECUTE  INFO  の2つが選択可能 |

X-Commandに指定可能なパラメータ

**表 3‑3　X-Commandに指定可能なパラメータ一覧**

|  |  |  |  |
| --- | --- | --- | --- |
| X-Command | 説明 | 対象画面 | メニューID |
| INFO | エクスポート可能なメニューの一覧を取得します。 | メニューエクスポート | 2100000211 |
| EXECUTE | エクスポートを実行します。 | メニューエクスポート | 2100000211 |

以下では、それぞれのX-comanndパラメータについての説明を行います。

### INFO

エクスポート可能なメニューの一覧を出力します。

・パラメータ

　指定するパラメータはありません。

　・レスポンス

　　返されるレスポンスには、JSON形式で格納されています。

|  |
| --- |
| {  "status": "SUCCEED",  "resultdata": {  "MENU\_LIST": {  "メニューグループID ": {  "menu\_group\_name": "メニューグループ名",  "menu": [  {  "menu\_id": "メニューID ",  "menu\_name": "メニュー名"  },  {  ・・・（中略）・・・  }  ]  },  "メニューグループID ": {  ・・・（中略）・・・  }  }  }  } |

表 3‑4　レスポンス項目一覧

|  |  |
| --- | --- |
| 項目名 | 備考 |
| メニューグループID | メニューグループIDをキーとしてメニューの**配列を構成する** |
| menu\_group\_name | メニューグループ名 |
| menu\_id | メニューID |
| menu\_name | メニュー名 |

### EXECUTE

対象のメニューを指定し、エクスポートを実行します。

・パラメータ

　　以下をJSON形式でcontentに指定してください。

**表 3‑5 メニューエクスポートパラメータ一覧**

|  |  |
| --- | --- |
| パラメータ名 | 設定値 |
| メニューグループID | メニューID |
| dp\_mode | モード  1:環境移行  2:時刻指定 |
| abolished\_type | 廃止情報  1:廃止を含む  2:廃止を除く |
| specified\_timestamp | 指定時刻  モードで2.時刻指定を指定するときのみ入力します。  YYYY-MM-DD H:i型で指定してください  例：2020-01-01 00:00 |

※メニューグループID、メニューIDは、INFOの返り値で取得したものです。

|  |
| --- |
| **例）JSON記述** |
| {  "2100000002": [  2100000202,  ・・・（中略）・・・  2100000222  ],  "2100000003": [  ・・・（中略）・・・  ],  “dp\_mode”: “2”,  “abolished\_type”: “1”,  “specified\_timestamp”:” 2020-01-01 00:00”  } |
|
|

・レスポンス

　　返されるレスポンスには、JSON形式で格納されています。項目について以下参照。

|  |
| --- |
| {  "status": "SUCCEED",  "resultdata": {  "TASK\_ID": "実行No ",  "RESULTCODE": "結果コード",  "RESULTINFO": "詳細情報"  }  } |
|
|

表 3‑6　レスポンス項目一覧

|  |  |
| --- | --- |
| 項目名 | 備考 |
| TASK\_ID | 作業No  作業Noで「エクスポート/インポート管理」メニューを検索することにより、実行状況を確認できます。 |
| RESULTCODE | コマンド実行の成否のコード  000：正常終了  002：実行不可 |
| RESULTINFO | 詳細情報 |

## メニューインポートを対象としたRestAPI

メニューエクスポートの操作をRestAPIで行うことができます。

利用可能な機能は、メニューグループ「エクスポート/インポート」の、「メニューインポート」メニューに相当する操作です。

表 3‑7 メニュー一覧

|  |  |  |
| --- | --- | --- |
| メニューグループ | メニュー名 | メニューID |
| エクスポート/インポート | メニューインポート | 2100000212 |

### リクエストの形式

下記の情報でHTTPリクエストを発行します。

・パス

https://<HostName>:<Port>/default/menu/07\_rest\_api\_ver1.php?no=メニューID

メニューIDは**「表 3‑9 X-Commandに指定可能なパラメータ一覧」**を参照してください**。**

・HTTPヘッダ

**表 3‑8　HTTPヘッダパラメータ一覧**

|  |  |
| --- | --- |
| HTTPヘッダ | 説明 |
| Content-Type | “application/json”を指定する。 |
| Authorization | ITAの認証要メニューにアクセスする場合は、  「ログインID」と「パスワード」\* を、半角コロン(:)で結合して、base64encodeをした値、を指定。 |
| X-Command | UPLOAD  EXECUTE  の2つが選択可能 |

X-Commandに指定可能なパラメータ

表 3‑9 X-Commandに指定可能なパラメータ一覧

|  |  |  |  |
| --- | --- | --- | --- |
| X-Command | 説明 | 対象画面 | メニューID |
| UPLOAD | エクスポートしたkymファイルのアップロードを実施し、インポート可能なメニューリストを出力します。 | メニューインポート | 2100000212 |
| EXECUTE | インポート対象のメニューを選択し、インポートを実施する。 | メニューインポート | 2100000212 |

以下では、それぞれのX-comanndパラメータについての説明を行います。

### UPLOAD

エクスポートされたファイルのアップロードを実行します。

ファイルはbase64encodeしたものをパラメータとして転送します。

・パラメータ

　以下をJSON形式でcontentに指定してください。

**表 3‑10　メニューインポートUPLOADパラメータ一覧**

|  |  |
| --- | --- |
| パラメータ名 | 設定値 |
| name | 対象のファイル名 |
| base64 | 対象のファイルをbase64encodeした値を指定 |

|  |
| --- |
| 1）UPLOAD　Json記述例 |
| {  "zipfile":{  "name":"ita\_exportdata\_20191224092830.kym",  "base64":"・・・中略・・・"  }  } |

・レスポンス

　　返されるレスポンスには、JSON形式で格納されています。

|  |
| --- |
| {  "status": ""SUCCEED",  "resultdata": {  "upload\_id": "アップロードID ",※  "data\_portability\_upload\_file\_name": "ファイル名",  “dp\_mode”: “1”,  “abolished\_type”: “1”,  "specified\_timestamp":null,  "IMPORT\_LIST": {  "メニューグループID ": {  "menu\_group\_name": "メニューグループ名",  "menu": [  {  "menu\_id": "メニューID ",  "menu\_name": "メニュー名"  }  ・・・（中略）・・・  ],  "メニューグループID ": {  ・・・（中略）・・・  ],  "RESULTCODE": "結果コード",  "RESULTINFO": "詳細情報"  }  }  }  } |
|
|

※「upload\_id」インポートの操作（EXECUTE）する際に使用します。

表 3‑11　レスポンス項目一覧

|  |  |
| --- | --- |
| 項目名 | 備考 |
| upload\_id | アップロード成功時に付与される値  EXECUTE時に使用 |
| data\_portability\_upload\_file\_name | ファイル名 |
| dp\_mode | モード  1:環境移行  2:時刻指定 |
| abolished\_type | 廃止情報  1:廃止を含む  2:廃止を除く |
| specified\_timestamp | 指定時刻  dp\_modeが1の場合はnullが設定されます。 |
| メニューグループID | メニューグループIDをキーとしてメニューの配列を構成する |
| menu\_group\_name | メニューグループ名 |
| menu\_id | メニューID |
| menu\_name | メニュー名 |
| RESULTCODE | コマンド実行の成否のコード  000：正常終了  002：実行不可 |
| RESULTINFO | 詳細情報 |

### EXECUTE

アップロードしたファイルを元に、インポート処理を実行します。

対象とする、メニューグループ、メニューID、インポート実行モードを指定できます。

・パラメータ

　　以下をJSON形式でcontentに指定してください。

**表 3‑12** **メニューインポートEXECUTEパラメータ**

|  |  |  |
| --- | --- | --- |
| パラメータ名 | 設定値 | 備考 |
| メニューグループID | メニューID |  |
| upload\_id |  | UPLOADの返り値で取得したものの先頭に”A\_”を付与した値 |
| data\_portability\_upload\_file\_name | ファイル名 |  |

|  |
| --- |
| 1）EXECUTE　Json記述例 |
| {  "2100070001": [  2100070001,  2100070002,  2100070003  ],  "2100020002": [  ・・・（中略）・・・  ],  "upload\_id": "A\_20191217090335772040239",※  "data\_portability\_upload\_file\_name": "ita\_exportdata\_20191213095733.kym"  } |

※UPLOADで取得した「upload\_id」の先頭に”A\_”を付与して使用します。

・レスポンス

　　返されるレスポンスには、JSON形式で格納されています。項目について以下参照。

|  |
| --- |
| {  "status": "SUCCEED",  "resultdata": {  "TASK\_ID": "メニューインポート実行の作業No",  "RESULTCODE": "結果コード",  "RESULTINFO": "詳細情報"  }  } |
|
|

表 3‑13　レスポンス項目一覧

|  |  |
| --- | --- |
| 項目名 | 備考 |
| TASK\_ID | 作業No  作業Noで「エクスポート/インポート管理」メニューを検索することにより、実行状況を確認できます。 |
| RESULTCODE | コマンド実行の成否のコード  000：正常終了  002：実行不可 |
| RESULTINFO | 詳細情報 |

# Symphony利用編

## Symphony登録作業を対象としたRestAPI

Symphonyの操作をRestAPIで行うことができます。

利用可能な機能は、メニューグループ「Symphony」の、「Symphonyクラス編集」メニューに相当する操作です。

表 4‑1 対象メニュー一覧

|  |  |  |
| --- | --- | --- |
| メニューグループ | メニュー名 | メニューID |
| Symphony | Symphonyクラス編集 | 2100000306 |

### リクエストの形式

下記の情報でHTTPリクエストを発行します。

・パス

https://<HostName>:<Port>/default/menu/07\_rest\_api\_ver1.php?no=メニューID

メニューIDは**「表 4‑3　X-Commandに指定可能なパラメータ一覧」**を参照してください**。**

・HTTPヘッダ

表 4‑2　HTTPヘッダパラメータ一覧

|  |  |
| --- | --- |
| HTTPヘッダ | 説明 |
| Content-Type | “application/json”を指定する。 |
| Authorization | ITAの認証要メニューにアクセスする場合は、  「ログインID」と「パスワード」\* を、半角コロン(:)で結合して、base64encodeをした値、を指定。 |
| X-Command | INFO  FILTER  FILTER\_DATAONLY  EDIT  の4つが選択可能 |

X-Commandに指定可能なパラメータ

表 4‑3　X-Commandに指定可能なパラメータ一覧

|  |  |  |  |
| --- | --- | --- | --- |
| X-Command | 説明 | 対象画面 | メニューID |
| INFO | Symphonyクラスの列情報のみを取得します。 | Symphonyクラス編集 | 2100000306 |
| FILTER  FILTER\_DATAONLY | Symphonyクラスのパラメータに一致したレコードの参照を行います。 | Symphonyクラス編集 | 2100000306 |
| EDIT | Symphonyクラスの登録を行います。 | Symphonyクラス編集 | 2100000306 |

以下では、それぞれのX-comanndパラメータについての説明を行います。

### INFO

Symphonyクラスの列情報を取得します。

※詳細は「標準REST機能の利用」-「INFO(X-Command)」を参照

### FILTER・FILTER\_DATAONLY

パラメータで指定した条件に合致したレコードの、列情報（列番号と列名）および、

通常ステータス（廃止または活性中）の全レコードの行数とレコード内容と列情報を取得します。

FILTER\_DATAONLYを指定した場合は列情報を省略します。

※詳細は「標準REST機能の利用」-「FILTER(X-Command)」を参照

### EDIT

Symphonyクラスの登録、更新、廃止、復活を行います。

・HTTPヘッダ

表 4‑4 HTTPヘッダパラメータ一覧

|  |  |
| --- | --- |
| HTTPヘッダ | 値 |
| Method | POST |
| X-Command | EDIT |

・パラメータ

1. 指定形式

各実行種別のパラメータ指定項目は以下の、パラメータ指定項目を参照してください。

項目番号7について、[更新]、[廃止]、[復活]指定時、

X-Command：FILTERで取得した、【更新用の最終更新日時】をセットしてください。

このデータによって、追い越し更新を防止しています。

【更新用の最終更新日時】は、“T\_“で始まっています。

**表 4‑5** **Symphonyクラスパラメータ一覧**

|  |  |  |
| --- | --- | --- |
| **項目番号** | **パラメータ名** | **備考** |
| 0 | 実行処理種別 | 登録/更新/廃止/復活 |
| 2 | SymphonyクラスID | 登録時は、空で実施。 |
| 3 | Symphony名称 |  |
| 4 | 説明 |  |
| 5 | 備考 |  |
| 7 | 更新用の最終更新日時 | T\_XXXXXXXXXXXXXXXXXXXX |
| 9 | Movement詳細 | Movement情報  詳細は以下、表 4‑6 |

**表 4‑6** **Movement詳細一覧**

|  |  |  |
| --- | --- | --- |
| **項目番号** | **パラメータ名** | **備考** |
| 0 | Orchestrator ID | オーケストレータのID  対応表は以下表 4‑8 |
| 1 | Movement ID | Movement のID  「Movement一覧」メニュー参照 |
| 2 | 一時停止 | OFF:空  ON:checkedValue |
| 3 | 説明 |  |
| 4 | オペレーションID(個別指定) | オペレーションのID  「投入オペレーション」メニューを参照。 |

**表 4‑7** **SymphonyクラスパラメータMovement詳細**

|  |
| --- |
| **パラメータ指定項目（登録/更新）** |
| "9": [  {  "0": "Orchestrator ID",  "1": "Movement ID",  "2": "一時停止(OFF:/ON:checkedValue)",  "3": "説明",  "4": "オペレーションID(個別指定)"  },  {  **・・・複数のMovement実行の場合、実行順に追加・・・・**  }  ] |
|
|
|
|
|

表 4‑8　オーケストレータのID対応表

|  |  |
| --- | --- |
| ID | ステータス |
| 3 | Ansible Legacy |
| 4 | Ansible Pioneer |
| 5 | Ansible Legacy Role |
| 10 | Terraform |

|  |
| --- |
| **パラメータ指定項目（登録/更新）** |
| {  "0": "実行処理種別：<登録 or 更新>",  "2": "SymphonyクラスID ",  "3": "Symphony名称",  "4": "説明",  "5": "備考",  "7": "更新用の最終更新日時",  "9": [  {  "0": "Orchestrator ID",  "1": "Movement ID",  "2": "一時停止(OFF:/ON:checkedValue)",  "3": "説明",  "4": "オペレーションID(個別指定)"  }  **・・・複数の場合、Movementの実行順で繰り返し追加・・・・**  ]  } |
|
|
|
|
|

|  |
| --- |
| **パラメータ指定項目（廃止/復活）** |
| {  "0": "実行処理種別：<廃止 or 復活>",  "2": "SymphonyクラスID",  "7": "更新用の最終更新日時"  } |
|
|
|

※Method：GET、X-Command：FILTERで取得した、【更新用の最終更新日時】をセット

してください。このデータによって、追い越し更新を防止しています。

【更新用の最終更新日時】は、“T\_“で始まっています。

|  |
| --- |
| **例）JSON記述例：複数の実行処理種別対象とした場合** |
| [  {  "0": "登録",  "2": "",  "3": "DEMO\_001\_20191224135448\_0",  "4": "demo\_001\_20191224135448\_0",  "7": "",  "9": [  {  "1": 3,  "2": 1,  "3": "checkedValue",  "4": "DEMO\_MOVE\_0",  "5": 1  },  {  "1": 3,  "2": 2,  "3": "",  "4": "DEMO\_MOVE\_1",  "5": ""  }  ]  },  {  "0": "更新",  "2": 1,  "3": "DEMO\_001\_20191224135448\_1",  "4": "demo\_001\_20191224135448\_1",  "7": "T\_20191224113132971799",  "9": [  {  "1": 3,  "2": 1,  "3": "",  "4": "DEMO\_MOVE\_0",  "5": 1  }  ]  },  {  "0": "廃止",  "2": 2,  "7": "T\_20191224135437197447"  },  {  "0": "復活",  "2": 4,  "7": "T\_20191224135449793941"  }  ] |
|
|

・レスポンス

登録・更新のRAW出力

表 4‑9 Keyパラメータ一覧

|  |  |  |
| --- | --- | --- |
| **key** | **値の型** |  |
| 0 | 文字列 | 結果コード(別表を参照) |
| 1 | 文字列 | 詳細コード(別表を参照) |
| 2 | 文字列 | SymphonyクラスID |
| 3 | 文字列 | エラーメッセージ |

各レコードの処理結果について、廃止・復活時のRAW出力は「標準REST機能の利用」-「EDIT(X-Command)」を参照。

## Symphony作業実行を対象としたRestAPI

Symphonyの操作をRestAPIで行うことができます。

利用可能な機能は、メニューグループ「Symphony」の、「Symphony作業実行」メニュー、「Symphony作業確認」メニューに相当する操作です。

表 4‑9　対象メニュー一覧

|  |  |  |
| --- | --- | --- |
| メニューグループ | メニュー名 | メニューID |
| Symphony | Symphony作業実行 | 2100000308 |
| Symphony作業確認 | 2100000309 |

### リクエストの形式

下記の情報でHTTPリクエストを発行します。

・パス

https://<HostName>:<Port>/default/menu/07\_rest\_api\_ver1.php?no=メニューID

メニューIDは**「表 4‑11　X-Commandに指定可能なパラメータ一覧」**を参照してください**。**

・HTTPヘッダ

表 4‑10 HTTPヘッダパラメータ一覧

|  |  |
| --- | --- |
| HTTPヘッダ | 説明 |
| Content-Type | “application/json”を指定する。 |
| Authorization | ITAの認証要メニューにアクセスする場合は、  「ログインID」と「パスワード」\* を、半角コロン(:)で結合して、base64encodeをした値、を指定。 |
| X-Command | EXECUTE  CANCEL  SCRAM  RELEASE  の4つが選択可能 |

X-Commandに指定可能なパラメータ

表 4‑11　X-Commandに指定可能なパラメータ一覧

|  |  |  |  |
| --- | --- | --- | --- |
| X-Command | 説明 | 対象画面 | メニューID |
| EXECUTE | Symphonyの作業実行を行います。 | Symphony作業実行 | 2100000308 |
| CANCEL | Symphonyの予約取り消しを行います。 | Symphony作業確認 | 2100000309 |
| SCRAM | Symphonyの緊急停止を行います。 | Symphony作業確認 | 2100000309 |
| RELEASE | Symphonyの一時停止ポイントの解除を行います。 | Symphony作業確認 | 2100000309 |

以下では、それぞれのX-comanndパラメータについての説明を行います。

### レスポンスの項目

以下では、それぞれのX-comannd実行時のレスポンス項目についての説明を行います。

表 4‑12　レスポンス項目一覧

|  |  |
| --- | --- |
| 項目名 | 備考 |
| SYMPHONY\_INSTANCE\_ID | SYMPHONYインスタンスに対する操作に使用 |
| MOVEMENT\_SEQ\_NO | RELEASE時のみ使用 |
| RESULTCODE | コマンド実行の成否のコード  000：正常終了  001：実行不可  002：予約取消不可  003：緊急停止不可  004：一時停止解除不可 |
| RESULTINFO | 詳細情報 |

### EXECUTE

Symphonyクラスとオペレーションを指定して、作業実行を行います。予約日時の指定や、Symphony

クラスに登録されているMovementごとに、スキップ、オペレーションIDの個別指定ができます。

・パラメータ

　以下をJSON形式でcontentに指定してください。

表 4‑13　オペレーションID　個別指定パラメータ一覧

|  |  |
| --- | --- |
| パラメータ名 | 設定値 |
| SYMPHONY\_CLASS\_NO | SymphonyクラスID |
| OPERATION\_ID | オペレーションID |
| PRESERVE\_DATETIME | 予約日時（YYYY/MM/DD ｔｔ:ｍｍ） |
| OPTION | スキップの有無、オペレーションIDの個別指定の配列 |

・OPTIONの指定

　　OPTIONには、配列でMovementごとに、スキップ、オペレーションIDの個別指定ができます。

・Movement要素の階層

├1（Movementの実行順番）

｜　　　　├SKIP　‐　YES or NO

｜ └OPERATION\_ID ‐ (個別指定するオペレーションID)

├2（Movementの実行順番）

｜ ├SKIP　‐　YES or NO

｜ └OPERATION\_ID ‐ (個別指定するオペレーションID)

・

・

|  |
| --- |
| 1）EXECUTE　Json記述例 |
| SymphonyクラスIDが1、オペレーションIDが1001、予約日時が2016/01/01 00:00の場合  さらに、1番目に実行されるMovementをスキップし、2番目に実行されるMovementのオペレーションIDに2001を指定  ▽Json形式で記述  {  "SYMPHONY\_CLASS\_NO": 1,  "OPERATION\_ID": 1001,  "PRESERVE\_DATETIME": "2016/01/0100:00",  "OPTION": {  "1": {  "SKIP": "YES"  },  "2": {  "OPERATION\_ID": 2001  }  }  } |

**図 6.1-1 EXECUTE　Json記述例**

　・レスポンス

　　返されるレスポンスには、JSON形式で格納されています。

|  |
| --- |
| {  "status": "実行の成否",  "resultdata": {  "SYMPHONY\_INSTANCE\_ID": "実行No ",※  "RESULTCODE": "結果コード",  "RESULTINFO": "詳細情報"  }  } |
|
|

※実行後のインスタンスを操作（INFO,CANCEL、SCRAM、RELEASE）する際に使用します。

### CANCEL

予約日時が登録されているSymphonyのインスタンスIDを指定して、予約実行をキャンセルします。

・パラメータ

　　以下をJSON形式でcontentに指定してください。

表 4‑14　 Symphony実行予約キャンセルパラメータ表

|  |  |
| --- | --- |
| パラメータ名 | 設定値 |
| SYMPHONY\_INSTANCE\_ID | SymphonyインスタンスID※ |

※EXECUTEの返り値で取得したものです。

・レスポンス

　　返されるレスポンスには、JSON形式で格納されています。項目について以下参照。

|  |
| --- |
| {  "status": "SUCCEED",  "resultdata": {  " SYMPHONY\_INSTANCE\_ID": " Symphony実行時のインスタンスのID ",  "RESULTCODE": "結果コード",  "RESULTINFO": "詳細情報"  }  } |
|
|

### SCRAM

実行されているSymphonyのインスタンスIDを指定して、緊急停止します。

・パラメータ

　　以下をJSON形式でcontentに指定してください。

表 4‑15 Symphony実行処理の緊急停止パラメータ表

|  |  |
| --- | --- |
| パラメータ名 | 設定値 |
| SYMPHONY\_INSTANCE\_ID | SymphonyインスタンスID※ |

※EXECUTEの返り値で取得したものです。

・レスポンス

　　返されるレスポンスには、JSON形式で格納されています。項目について以下参照。

|  |
| --- |
| {  "status": "SUCCEED",  "resultdata": {  " SYMPHONY\_INSTANCE\_ID": " Symphony実行時のインスタンスのID ",  "RESULTCODE": "結果コード",  "RESULTINFO": "詳細情報"  }  } |
|
|

### RELEASE

SymphonyのインスタンスIDとMovementの順番を指定して、一時停止が設定されているポイントを解除します。

・パラメータ

　　以下をJSON形式でcontentに指定してください。

表 4‑16 Symphony実行処理の一時停止解除パラメータ表

|  |  |
| --- | --- |
| パラメータ名 | 設定値 |
| SYMPHONY\_INSTANCE\_ID | SymphonyインスタンスID※ |
| MOVEMENT\_SEQ\_NO | 何番目のMovementか |

※EXECUTEの返り値で取得したものです。

・レスポンス

　　返されるレスポンスには、JSON形式で格納されています。項目について以下参照。

|  |
| --- |
| {  "status": "SUCCEED",  "resultdata": {  "SYMPHONY\_INSTANCE\_ID": "Symphony実行時のインスタンスのID",  "MOVEMENT\_SEQ\_NO": "実行したMovementのSymphonyクラス内の順番",  "RESULTCODE": "結果コード",  "RESULTINFO": "詳細情報"  }  } |
|
|

## Symphony作業確認を対象としたRestAPI

Symphonyの操作をRestAPIで行うことができます。

利用可能な機能は、メニューグループ「Symphony」の、「Symphony作業確認」メニューに相当する操作です。

表 4‑17　対象メニュー一覧

|  |  |  |
| --- | --- | --- |
| メニューグループ | メニュー名 | メニューID |
| Symphony | Symphony作業確認 | 2100000309 |

### リクエストの形式

下記の情報でHTTPリクエストを発行します。

・パス

https://<HostName>:<Port>/default/menu/07\_rest\_api\_ver1.php?no=メニューID

メニューIDは**「表 4‑19　X-Commandに指定可能なパラメータ一覧」**を参照してください**。**

・HTTPヘッダ

表 4‑18 HTTPヘッダパラメータ一覧

|  |  |
| --- | --- |
| HTTPヘッダ | 説明 |
| Content-Type | “application/json”を指定する。 |
| Authorization | ITAの認証要メニューにアクセスする場合は、  「ログインID」と「パスワード」\* を、半角コロン(:)で結合して、base64encodeをした値、を指定。 |
| X-Command | INFO  の1つが選択可能 |

X-Commandに指定可能なパラメータ

表 4‑19　X-Commandに指定可能なパラメータ一覧

|  |  |  |  |
| --- | --- | --- | --- |
| X-Command | 説明 | 対象画面 | メニューID |
| INFO | Symphonyの状態確認をし、ステータスを返します。 | Symphony作業確認 | 2100000309 |

以下では、それぞれのX-comanndパラメータについての説明を行います。

### レスポンスの項目

以下では、それぞれのX-comannd実行時のレスポンス項目についての説明を行います。

表 4‑20　レスポンス項目一覧

|  |  |
| --- | --- |
| 項目名 | 備考 |
| SYMPHONY\_INSTANCE\_ID | SYMPHONYインスタンスに対する操作に使用 |
| RESULTCODE | コマンド実行の成否のコード  000：正常終了 |
| RESULTINFO | 詳細情報 |

### INFO

Symphony実行時のインスタンスIDを指定して、実行時の情報を取得します。

・パラメータ

　　以下をJSON形式でcontentに指定してください。

表 4‑21　Symphony実行情報取得パラメータ表

|  |  |
| --- | --- |
| パラメータ名 | 設定値 |
| SYMPHONY\_INSTANCE\_ID | SymphonyインスタンスID※ |

※EXECUTEの返り値で取得したものです。

・レスポンス

　　返されるレスポンスには、JSON形式で格納されています。

|  |
| --- |
| {  "status": "SUCCEED",  "resultdata": {  "SYMPHONY\_CLASS\_ID": "1",  "SYMPHONY\_INSTANCE\_INFO": {  "SYMPHONY\_INSTANCE\_ID": 1,  ・・・（中略）項目は以下①参照・・・  "FOCUS\_MOVEMENT": 1  },  "MOVEMENTS": [  {  "CLASS\_ITEM": {  "ORCHESTRATOR\_ID": "3",  ・・・（中略）項目は以下②参照・・・  "NEXT\_PENDING": "checkedValue"  },  "INS\_ITEM": {  "STATUS": "11",  ・・・（中略）項目は以下③参照・・・  "OPERATION\_NAME": null  }  }  　　　　　　・・・・  ],  "RESULTCODE": "000",  "RESULTINFO": ""  }  } |
|
|

①SYMPHONY\_INSTANCE\_INFOに格納されるSymphonyインスタンスの情報配列

表 4‑22　インスタンス配列表

|  |  |
| --- | --- |
| キー | 内容 |
| SYMPHONY\_INSTANCE\_ID | SymphonyインスタンスID |
| I\_SYMPHONY\_CLASS\_NO | このインスタンスの元クラスのID |
| I\_SYMPHONY\_NAME | このインスタンスの元クラスの名前 |
| I\_DESCRIPTION | このインスタンスの元クラスの説明 |
| STATUS\_ID | 実行時ステータス　詳細は以下　表 4‑25 |
| ABORT\_EXECUTE\_FLAG | 緊急停止発令フラグ　未発令：１　発令済み：２ |
| OPERATION\_NO\_UAPK | 登録オペレーションNO |
| OPERATION\_NO\_IDBH | 登録オペレーションID |
| OPERATION\_NAME | 登録オペレーション名 |
| TIME\_BOOK | 予約日時 |
| TIME\_START | 開始日時 |
| TIME\_END | 終了日時 |
| MOVEMENT\_LENGTH | 登録Movementの数 |
| FOCUS\_MOVEMENT | 現在のMovementは何番目か |

②CLASS\_ITEMに格納されるMovementのクラス情報

表 4‑23　Movementクラス情報表

|  |  |
| --- | --- |
| キー | 内容 |
| ORCHESTRATOR\_ID | オーケストレータのID　　対応表は以下　表 4‑26 |
| PATTERN\_ID | MovementのID |
| PATTERN\_NAME | Movementの名前 |
| THEME\_COLOR | <画面用>Web画面で設定時の○アイコンの色 |
| MOVEMENT\_SEQ | Symphonyクラスの中で何番目か |
| DESCRIPTION | Symphonyクラス編集画面で入力したコメント |
| NEXT\_PENDING | 一時停止が設定されている：checkedValue |

③INS\_ITEMに格納されるMovementのインスタンス情報

表 4‑24　Movementインスタンス情報表

|  |  |
| --- | --- |
| キー | 内容 |
| STATUS | 実行時ステータス　詳細は以下　表 4‑27 |
| RELEASED | 一時停止が設定されている:1  一時停止解除された：2 |
| EXECUTION\_NO | MovementインスタンスID |
| JUMP | <画面用>遷移先URL |
| ABORT\_RECEPTED | 緊急停止を　１：受け付けていない　２：受付済み |
| SKIP | スキップが設定されている　：1 |
| TIME\_START | 開始日時 |
| TIME\_END | 終了日時 |
| OPERATION\_ID | 個別指定されたオペレーションID |
| OPERATION\_NAME | 個別指定されたオペレーション名 |

表 4‑25　Symphonyインスタンスの実行時ステータスID対応表

|  |  |
| --- | --- |
| ID | ステータス |
| 1 | 未実行 |
| 2 | 未実行(予約) |
| 3 | 実行中 |
| 4 | 実行中(遅延) |
| 5 | 正常終了 |
| 6 | 緊急停止 |
| 7 | 異常終了 |
| 8 | 想定外エラー |
| 9 | 予約取消 |

表 4‑26　オーケストレータのID対応表

|  |  |
| --- | --- |
| ID | ステータス |
| 3 | Ansible Legacy |
| 4 | Ansible Pioneer |
| 5 | Ansible Legacy Role |
| 10 | Terraform |

表 4‑27 Movementインスタンスの実行時ステータスID対応表

|  |  |
| --- | --- |
| ID | ステータス |
| 1 | 未実行 |
| 2 | 準備中 |
| 3 | 実行中 |
| 4 | 実行中(遅延) |
| 5 | 実行完了 |
| 6 | 異常終了 |
| 7 | 緊急停止 |
| 8 | 保留中 |
| 9 | 正常終了 |
| 10 | 準備エラー |
| 11 | 想定外エラー |
| 12 | Skip完了 |
| 13 | Skip後保留中 |
| 14 | Skip終了 |

# Conductor利用編

## Conductor登録作業を対象としたRestAPI

Conductorの操作をRestAPIで行うことができます。

利用可能な機能は、メニューグループ「Conductor」の、「Conductorクラス編集」メニューに相当する操作です。

表 5‑1 対象メニュー一覧

|  |  |  |
| --- | --- | --- |
| メニューグループ | メニュー名 | メニューID |
| Conductor | Conductorクラス編集 | 2100180003 |

### リクエストの形式

下記の情報でHTTPリクエストを発行します。

・パス

https://<HostName>:<Port>/default/menu/07\_rest\_api\_ver1.php?no=メニューID

メニューIDは**「表 4‑3　X-Commandに指定可能なパラメータ一覧」**を参照してください**。**

・HTTPヘッダ

表 5‑2　HTTPヘッダパラメータ一覧

|  |  |
| --- | --- |
| HTTPヘッダ | 説明 |
| Content-Type | “application/json”を指定する。 |
| Authorization | ITAの認証要メニューにアクセスする場合は、  「ログインID」と「パスワード」\* を、半角コロン(:)で結合して、base64encodeをした値、を指定。 |
| X-Command | INFO  FILTER  FILTER\_DATAOLY  EDIT  の4つが選択可能 |

X-Commandに指定可能なパラメータ

表 5‑3　X-Commandに指定可能なパラメータ一覧

|  |  |  |  |
| --- | --- | --- | --- |
| X-Command | 説明 | 対象画面 | メニューID |
| INFO | Conductorクラスの列情報のみを取得します。 | Conductorクラス編集 | 2100180003 |
| FILTER FILTER\_DATAONLY | Conductorクラスのパラメータに一致したレコードの参照を行います。 | Conductorクラス編集 | 2100180003 |
| EDIT | Conductorクラスの登録を行います。 | Conductorクラス編集 | 2100180003 |

以下では、それぞれのX-comanndパラメータについての説明を行います。

### INFO

Conductorクラスの列情報を取得します。

※詳細は「標準REST機能の利用」-「INFO(X-Command)」を参照

|  |
| --- |
| **例）INFO取得例** |
| {…  "INFO":{  "edittype": "実行処理種別",  "conductor": {  "conductor\_name":"Conductor名称",  "id":" ConductorクラスID",  "note":"説明",  "LUT4U":"更新用の最終更新日時",  "ACCESS\_AUTH":"アクセス権/アクセス許可ロール"  },  "config": {  "editorVersion":"Editor情報",  "nodeNumber":" Node数",  "terminalNumber":"Terminal数",  "edgeNumber":"Edge数"  },  "node": {  "h":"高さ",  "id":" Node class id",  "terminal":{  "case":"Case no",  "edge":" 紐付Line",  "id":"Terminal class id",  "targetNode":"紐付Node"  "type":"種別",  "condition":"状態",  "x":"横",  "y":"縦"  },  "type":"種別",  "PATTERN\_ID":"Pattern id",  "ORCHESTRATOR\_ID":"Orchestrator id",  "Name":" Movement名",  "CALL\_CONDUCTOR\_ID":"ConductorクラスID",  "CONDUCTOR\_NAME":"Conductor名称",  "CALL\_SYMPHONY\_ID":"SymphonyクラスID",  "SYMPHONY\_NAME":"Symphony名称",  "OPERATION\_NO\_IDBH:" オペレーションID",  "SKIP\_FLAG":"Skip flag",  "OPERATION\_NAME":"オペレーション名",  "note":"説明",  "w":"幅",  "x":"横",  "y":"縦"  },  "line": {  "id":"Line id",  "type":"種別",  "inNode":"内側接続Node",  "outTerminal":"外側接続Terminal "  "inTerminal":"内側接続Terminal ",  "outNode ":"外側接続Node "  }  }  } |
|
|

各項目キーの内容は表5-4～5.9参照

**表 5‑4** **Conductorクラスパラメータ一覧**

|  |  |  |
| --- | --- | --- |
| **項目キー** | **パラメータ名** | **備考** |
| edittype | 実行処理種別 | 登録/更新/廃止/復活 |
| conductor | Conductor詳細 | Conductor情報  詳細は以下、表5-5 |
| config | Config名称 | Config情報  詳細は以下、表5-6 |
| node | Node詳細 | Node情報  詳細は以下、表5-7 |
| terminal | Terminal詳細 | Terminal情報  詳細は以下、表5-8 |
| line | Line詳細 | Line情報  詳細は以下、表5-9 |

**表 5‑5** **Conductor詳細一覧**

|  |  |  |
| --- | --- | --- |
| **項目キー** | **パラメータ名** | **備考** |
| conductor\_name | Conductor名称 |  |
| id | ConductorクラスID | 登録時は、空で実施。 |
| note | 説明 |  |
| LUT4U | 更新用の最終更新日時 | T\_XXXXXXXXXXXXXXXXXXXX |
| ACCESS\_AUTH | アクセス権/アクセス許可ロール |  |

**表 5‑6** **Config詳細一覧**

|  |  |  |
| --- | --- | --- |
| **項目キー** | **パラメータ名** | **備考** |
| editorVersion | Editor情報 |  |
| nodeNumber | Node数 |  |
| terminalNumber | Terminal数 |  |
| edgeNumber | Edge数 |  |

**表 5‑7** **Node詳細一覧**

|  |  |  |
| --- | --- | --- |
| **項目キー** | **パラメータ名** | **備考** |
| h | 高さ |  |
| id | Node class id |  |
| type | 種別 | start/end/movement/call/call\_s/pause/ conditional-branch/ parallel-branch/ parallel-merge |
| PATTERN\_ID | Pattern id |  |
| ORCHESTRATOR\_ID | Orchestrator id | オーケストレータのID  対応表は以下表 4‑8 |
| Name | Movement名 |  |
| CALL\_CONDUCTOR\_ID | ConductorクラスID |  |
| CONDUCTOR\_NAME | Conductor名称 |  |
| CALL\_SYMPHONY\_ID | SymphonyクラスID |  |
| SYMPHONY\_NAME | Symphony名称 |  |
| OPERATION\_NO\_IDBH | オペレーションID |  |
| SKIP\_FLAG | Skip flag |  |
| OPERATION\_NAME | オペレーション名 |  |
| note | 説明 |  |
| w | 幅 |  |
| x | 横 |  |
| y | 縦 |  |

**表 5‑8** **Terminal詳細一覧**

|  |  |  |
| --- | --- | --- |
| **項目キー** | **パラメータ名** | **備考** |
| case | Case no |  |
| edge | Node class id |  |
| id | Terminal class id |  |
| targetNode | 紐付Node |  |
| type | 種別 | in/out |
| condition | 状態 |  |
| x | 横 |  |
| y | 縦 |  |

**表 5‑9** **Line詳細一覧**

|  |  |  |
| --- | --- | --- |
| **項目キー** | **パラメータ名** | **備考** |
| id | Line id |  |
| type | 種別 | egde |
| inNode | 内側接続Node |  |
| outTerminal | 外側接続Terminal |  |
| outTerminal | 内側接続Terminal |  |
| outNode | 外側接続Node |  |

### FILTER・FILTER\_DATAONLY

パラメータで指定した条件に合致したレコードの、列情報（列番号と列名）および、

通常ステータス（廃止または活性中）の全レコードの行数とレコード内容と列情報を取得します。

FILTER\_DATAONLYを指定した場合は列情報を省略します。

※詳細は「標準REST機能の利用」-「FILTER(X-Command)」を参照

取得列情報は表5-4～5.9参照

表 5‑10　パラメータで指定可能な項目

|  |  |
| --- | --- |
| 項目キー | 対象項目 |
| conductor\_name | Conductor名称 |
| id | ConductorクラスID |
| note | 説明 |
| ACCESS\_AUTH | アクセス権/アクセス許可ロール |

例） ConductorクラスID が1のレコードを取得したい場合  
{"id":{"LIST":["1"]}}

Conductorクラス名称にConductorが含まれるレコードを取得したい場合

{"conductor\_name":{"NORMAL":"Conductor"}}

ConductorクラスIDが1～3のレコードを取得したい場合

{"id":{"RANGE":{"START":"1","END":"3"}}}

|  |
| --- |
| **例）FILTER取得例** |
| {…  {  "edittype": "実行処理種別",  "disuse": "廃止",  "conductor": {  "conductor\_name":"Conductor名称",  "id":"ConductorクラスID",  "note":"説明",  "LUT4U":"更新用の最終更新日時",  "ACCESS\_AUTH":"アクセス権/アクセス許可ロール"  },  "config": {  "editorVersion":"Editor情報",  "nodeNumber":" Node数",  "terminalNumber":"Terminal数",  "edgeNumber":"Edge数"  },  "node": {  "h":"高さ",  "id":" Node class id",  "terminal":{  "case":"Case no",  "edge":" 紐付Line",  "id":"Terminal class id",  "targetNode":"紐付Node"  "type":"種別",  "condition":"状態",  "x":"横",  "y":"縦"  },  "type":"種別",  "PATTERN\_ID":"Pattern id",  "ORCHESTRATOR\_ID":"Orchestrator id",  "Name":" Movement名",  "CALL\_CONDUCTOR\_ID":"ConductorクラスID",  "CONDUCTOR\_NAME":"Conductor名称",  "CALL\_SYMPHONY\_ID":"SymphonyクラスID",  "SYMPHONY\_NAME":"Symphony名称",  "OPERATION\_NO\_IDBH:" オペレーションID",  "SKIP\_FLAG":"Skip flag",  "OPERATION\_NAME":"オペレーション名",  "note":"説明",  "w":"幅",  "x":"横",  "y":"縦"  },  "line": {  "id":"Line id",  "type":"種別",  "inNode":"内側接続Node",  "outTerminal":"外側接続Terminal "  "inTerminal":"内側接続Terminal ",  "outNode ":"外側接続Node "  }  },  1:{  "edittype":null,  "disuse":"",  "conductor":{  conductor\_name:Conductor\_002,  id:7,  note:null,  LUT4U:T\_20210629121745463936,  ACCESS\_AUTH:""  },  "config:{  "editorVersion":"1.0.2",  nodeNumber:4,  terminalNumber:5,  edgeNumber:3  },  node-1:{  "h":"58",  "id":"node-1",  "terminal":{  "terminal-1":{  "case":"",  "edge":"line-1",  "id":"terminal-1",  "targetNode":"node-3",  "type":"out",  "condition":[],  "x":"7666",  "y":"8000"  }  }  "type":"start",  "PATTERN\_ID":"",  "ORCHESTRATOR\_ID":"",  "Name":"",  "CALL\_CONDUCTOR\_ID":"",  "CONDUCTOR\_NAME":"",  "CALL\_SYMPHONY\_ID":"",  "SYMPHONY\_NAME":"",  "OPERATION\_NO\_IDBH":"",  "SKIP\_FLAG":"",  "OPERATION\_NAME":"",  "note":null,  "w":"198",  "x":"7485",  "y":"7971"  },  node-2:{  "h":"58",  "id":"node-2",  "terminal":{  "terminal-2":{  "case":"",  "edge":"line-2",  "id":"terminal-2",  "targetNode":"node-3",  "type":"in",  "condition":[],  "x":"8334",  "y":"8000"  }  }  "type":"end",  "PATTERN\_ID":"",  "ORCHESTRATOR\_ID":"",  "Name":"",  "CALL\_CONDUCTOR\_ID":"",  "CONDUCTOR\_NAME":"",  "CALL\_SYMPHONY\_ID":"",  "SYMPHONY\_NAME":"",  "OPERATION\_NO\_IDBH":"",  "SKIP\_FLAG":"",  "OPERATION\_NAME":"",  "note":null,  "w":"198",  "x":"8317",  "y":"7971"  },  node-3:{  "h":"58",  "id":"node-3",  "terminal":{  "terminal-3":{  "case":"",  "edge":"line-1",  "id":"terminal-3",  "targetNode":"node-1",  "type":"in",  "condition":[],  "x":"7888",  "y":"8000"  },  "terminal-4":{  "case":"",  "edge":"line-2",  "id":"terminal-4",  "targetNode":"node-2",  "type":"out",  "condition":[],  "x":"8099",  "y":"8000"  }  }  "type":"movement",  "PATTERN\_ID":"1",  "ORCHESTRATOR\_ID":"3",  "Name":"move1",  "CALL\_CONDUCTOR\_ID":"",  "CONDUCTOR\_NAME":"",  "CALL\_SYMPHONY\_ID":"",  "SYMPHONY\_NAME":"",  "OPERATION\_NO\_IDBH":"",  "SKIP\_FLAG":"0",  "OPERATION\_NAME":"",  "note":null,  "w":"245",  "x":"7871",  "y":"7978"  },  line-1:{  "id":"line-1",  "type":"egde",  "inNode":"node-3",  "outTerminal":"terminal-1",  "inTerminal":"terminal-3",  "outNode":"node-1"  },  line-2:{  "id":"line-2",  "type":"egde",  "inTerminal":"terminal-2",  "outNode":"node-3",  "inNode":"node-2",  "outTerminal":"terminal-4"  }  }  } |
|
|

NodeのORCHESTRATOR\_IDについては以下の表を参照

表 5‑11　 ORCHESTRATOR\_ID(オーケストレータID)対応表

|  |  |
| --- | --- |
| ID | ステータス |
| 3 | Ansible Legacy |
| 4 | Ansible Pioneer |
| 5 | Ansible Legacy Role |
| 10 | Terraform |

### EDIT

Conductorクラスの登録、更新、廃止、復活を行います。

・HTTPヘッダ

表 5‑12 HTTPヘッダパラメータ一覧

|  |  |
| --- | --- |
| HTTPヘッダ | 値 |
| Method | POST |
| X-Command | EDIT |

・パラメータ

1. 指定形式

各実行種別のパラメータ指定項目は以下の、パラメータ指定項目を参照してください。

項目キーLUT4Uについては、[更新]、[廃止]、[復活]指定時、

X-Command：FILTERで取得した、【更新用の最終更新日時】をセットしてください。

このデータによって、追い越し更新を防止しています。

【更新用の最終更新日時】は、“T\_“で始まっています。

|  |
| --- |
| **パラメータ指定項目（登録/更新）** |
| {  "edittype": "実行処理種別：<登録 or 更新>",  "conductor": {  "conductor\_name":"Conductor名称",  "id":" ConductorクラスID",  "note":"説明",  "LUT4U":"更新用の最終更新日時",  "ACCESS\_AUTH":"アクセス権/アクセス許可ロール"  },  "config": {  "editorVersion":"Editor情報",  "nodeNumber":" Node数",  "terminalNumber":"Terminal数",  "edgeNumber":"Edge数"  },  "node-1": {  "h":"高さ",  "id":" Node class id",  "terminal":{  "terminal-1": {  "case":"Case no",  "edge":" 紐付Line",  "id":"Terminal class id",  "targetNode":"紐付Node"  "type":"種別",  "condition":"状態",  "x":"横",  "y":"縦"  },  **・・・Terminal数分繰り返し追加(terminal-1, terminal-2… terminal-n)・・・・**  "type":"種別",  "PATTERN\_ID":"Pattern id",  "ORCHESTRATOR\_ID":"Orchestrator id",  "Name":" Movement名",  "CALL\_CONDUCTOR\_ID":"ConductorクラスID",  "CONDUCTOR\_NAME":"Conductor名称",  "CALL\_SYMPHONY\_ID":"SymphonyクラスID",  "SYMPHONY\_NAME":"Symphony名称",  "OPERATION\_NO\_IDBH:" オペレーションID",  "SKIP\_FLAG":"Skip flag",  "OPERATION\_NAME":"オペレーション名",  "note":"説明",  "w":"幅",  "x":"横",  "y":"縦"  },  **・・・Node数分繰り返し追加(node-1,node-2…node-n)・・・・**  "line-1": {  "id":"Line id",  "type":"種別",  "inNode":"内側接続Node",  "outTerminal":"外側接続Terminal "  "inTerminal":"内側接続Terminal ",  "outNode ":"外側接続Node "  },  **・・・Line数分繰り返し追加(line-1,line-2…line-n)・・・・**  } |
|
|
|
|
|

|  |
| --- |
| **パラメータ指定項目（廃止/復活）** |
| {  "edittype": "実行処理種別：<廃止 or 復活>",  "conductor": {  "id":" ConductorクラスID",  "LUT4U":"更新用の最終更新日時"  },  } |
|
|
|

※Method：GET、X-Command：FILTERで取得した、【更新用の最終更新日時】をセット

してください。このデータによって、追い越し更新を防止しています。

【更新用の最終更新日時】は、“T\_“で始まっています。

|  |
| --- |
| **例）JSON記述例：複数の実行処理種別対象とした場合** |
| {"0":{  "edittype": "登録",  "conductor": {  "conductor\_name":"Conductor\_001",  "note":null,  "ACCESS\_AUTH":null  },  "config": {  "editorVersion":"1.0.2",  "nodeNumber":4,  "terminalNumber":5,  "edgeNumber":3  },  "node-1": {  "h":58,  "id":" node-1",  "terminal":{  "terminal-1": {  "edge":" line-1",  "id":" terminal-1",  "targetNode":" node-3"  "type":"out",  "x":"7666",  "y":"8000"  },  "type":"start",  "note":null,  "w":198,  "x":7485,  "y":7971  },  "node-2": {  "h":58,  "id":" node-2",  "terminal":{  "terminal-2": {  "edge":" line-2",  "id":" terminal-2",  "targetNode":" node-3"  "type":"in",  "x":"8334",  "y":"8000"  },  "type":"end",  "note":null,  "w":198,  "x":8317,  "y":7971  },  "node-3": {  "h":58,  "id":" node-3",  "terminal":{  "terminal-3": {  "edge":" line-1",  "id":" terminal-3",  "targetNode":" node-1"  "type":"in",  "x":"7888",  "y":"8000"  },  "terminal-4": {  "edge":" line-2",  "id":" terminal-4",  "targetNode":" node-2"  "type":"out",  "x":"8099",  "y":"8000"  },  "type":"movement",  "PATTERN\_ID":"1",  "ORCHESTRATOR\_ID ":"3",  "Name ":"move1",  "OPERATION\_NO\_IDBH":null,  "SKIP\_FLAG ":"0",  "OPERATION\_NAME":"",  "note":null,  "w":198,  "x":8317,  "y":7971  },  "1":{  "edittype": "廃止",  "conductor": {  "id":"2",  "LUT4U": "T\_20191224135449793941"  }  }  } |
|
|

また、WebのConductorクラス編集画面からjsonファイルを保存して更新、登録、廃止、復活に流用することができます。



・レスポンス

登録・更新のRAW出力

表 5‑13 Keyパラメータ一覧

|  |  |  |
| --- | --- | --- |
| **key** | **値の型** |  |
| 0 | 文字列 | 結果コード(別表を参照) |
| 1 | 文字列 | 詳細コード(別表を参照) |
| 2 | 文字列 | ConnductorクラスID |
| 3 | 文字列 | エラーメッセージ |

各レコードの処理結果について、廃止・復活時のRAW出力は「標準REST機能の利用」-「EDIT(X-Command)」を参照。

## Conductor作業実行を対象としたRestAPI

Conductorの操作をRestAPIで行うことができます。

利用可能な機能は、メニューグループ「Conductor」の、「Conductor作業実行」メニュー、「Conductor作業確認」メニューに相当する操作です。

表 5‑14　対象メニュー一覧

|  |  |  |
| --- | --- | --- |
| メニューグループ | メニュー名 | メニューID |
| Conductor | Conductor作業実行 | 2100180004 |
| Conductor作業確認 | 2100180005 |

### リクエストの形式

下記の情報でHTTPリクエストを発行します。

・パス

https://<HostName>:<Port>/default/menu/07\_rest\_api\_ver1.php?no=メニューID

メニューIDは**「表 5‑16　X-Commandに指定可能なパラメータ一覧」**を参照してください**。**

・HTTPヘッダ

表 5‑15 HTTPヘッダパラメータ一覧

|  |  |
| --- | --- |
| HTTPヘッダ | 説明 |
| Content-Type | “application/json”を指定する。 |
| Authorization | ITAの認証要メニューにアクセスする場合は、  「ログインID」と「パスワード」\* を、半角コロン(:)で結合して、base64encodeをした値、を指定。 |
| X-Command | EXECUTE  CANCEL  SCRAM  RELEASE  の4つが選択可能 |

表 5‑16　X-Commandに指定可能なパラメータ一覧

|  |  |  |  |
| --- | --- | --- | --- |
| X-Command | 説明 | 対象画面 | メニューID |
| EXECUTE | Conductorの作業実行を行います。 | Conductor作業実行 | 2100180004 |
| CANCEL | Conductorの予約取り消しを行います。 | Conductor作業確認 | 2100180005 |
| SCRAM | Conductorの緊急停止を行います。 | Conductor作業確認 | 2100180005 |
| RELEASE | Conductorの一時停止の解除を行います。 | Conductor作業確認 | 2100180005 |

以下では、それぞれのX-comanndパラメータについての説明を行います。

### レスポンスの項目

以下では、それぞれのX-comannd実行時のレスポンス項目についての説明を行います。

表 5‑17　レスポンス項目一覧

|  |  |
| --- | --- |
| 項目名 | 備考 |
| CONDUCTOR\_INSTANCE\_ID | CONDUCTORインスタンスに対する操作に使用 |
| NODE\_INSTANCE\_NO | RELEASE時のみ使用 |
| RESULTCODE | コマンド実行の成否のコード  000：正常終了  001：実行不可  002：予約取消不可  003：緊急停止不可  004：一時停止解除不可 |
| RESULTINFO | 詳細情報 |

### EXECUTE

Conductorクラスとオペレーションを指定して、作業実行を行います。予約日時の指定や、Conductor

クラスに登録されているMovementごとに、スキップ、オペレーションIDの個別指定ができます。

・パラメータ

　以下をJSON形式でcontentに指定してください。

表 5‑18　オペレーションID　個別指定パラメータ一覧

|  |  |
| --- | --- |
| パラメータ名 | 設定値 |
| CONDUCTOR\_CLASS\_NO | ConductorクラスID |
| OPERATION\_ID | オペレーションID |
| PRESERVE\_DATETIME | 予約日時（YYYY/MM/DD ｔｔ:ｍｍ） |

|  |
| --- |
| 1）EXECUTE　Json記述例 |
| ConductorクラスIDが1、オペレーションIDが1001、予約日時が2016/01/01 00:00の場合  ▽Json形式で記述  {  "CONDUCTOR\_CLASS\_NO": 1,  "OPERATION\_ID": 1001,  "PRESERVE\_DATETIME": "2016/01/01 00:00",  } |

**図 6.1-1 EXECUTE　Json記述例**

　・レスポンス

　　返されるレスポンスには、JSON形式で格納されています。

|  |
| --- |
| {  "status": "実行の成否",  "resultdata": {  "CONDUCTOR\_INSTANCE\_ID": "実行No ",※  "RESULTCODE": "結果コード",  "RESULTINFO": "詳細情報"  }  } |
|
|

※実行後のインスタンスを操作（INFO,CANCEL、SCRAM、RELEASE）する際に使用します。

### CANCEL

予約日時が登録されているConductorのインスタンスIDを指定して、予約実行をキャンセルします。

・パラメータ

　　以下をJSON形式でcontentに指定してください。

表 5‑19　 Conductor実行予約キャンセルパラメータ表

|  |  |
| --- | --- |
| パラメータ名 | 設定値 |
| CONDUCTOR\_INSTANCE\_ID | ConductorインスタンスID※ |

※EXECUTEの返り値で取得したものです。

・レスポンス

　　返されるレスポンスには、JSON形式で格納されています。項目について以下参照。

|  |
| --- |
| {  "status": "SUCCEED",  "resultdata": {  " CONDUCTOR\_INSTANCE\_ID": " Conductor実行時のインスタンスのID ",  "RESULTCODE": "結果コード",  "RESULTINFO": "詳細情報"  }  } |
|
|

### SCRAM

実行されているConductorのインスタンスIDを指定して、緊急停止します。

・パラメータ

　　以下をJSON形式でcontentに指定してください。

表 5‑20 Conductor実行処理の緊急停止パラメータ表

|  |  |
| --- | --- |
| パラメータ名 | 設定値 |
| CONDUCTOR\_INSTANCE\_ID | ConductorインスタンスID※ |

※EXECUTEの返り値で取得したものです。

・レスポンス

　　返されるレスポンスには、JSON形式で格納されています。項目について以下参照。

|  |
| --- |
| {  "status": "SUCCEED",  "resultdata": {  " CONDUCTOR\_INSTANCE\_ID": " Conductor実行時のインスタンスのID ",  "RESULTCODE": "結果コード",  "RESULTINFO": "詳細情報"  }  } |
|
|

### RELEASE

ConductorのインスタンスIDとNodeインスタンスIDを指定して、一時停止が設定されているポイントを解除します。

・パラメータ

　　以下をJSON形式でcontentに指定してください。

表 5‑21 Conductor実行処理の一時停止解除パラメータ表

|  |  |
| --- | --- |
| パラメータ名 | 設定値 |
| CONDUCTOR\_INSTANCE\_ID | ConductorインスタンスID※1 |
| NODE\_INSTANCE\_ID | NodeインスタンスID※2 |

※1 EXECUTEの返り値で取得したものです。

※2 NodeのインスタンスIDの取得について、「 5.3 Conductor作業確認を対象としたRestAPI」を参照。

・レスポンス

　　返されるレスポンスには、JSON形式で格納されています。項目について以下参照。

|  |
| --- |
| {  "status": "SUCCEED",  "resultdata": {  "CONDUCTOR\_INSTANCE\_ID": "Conductor実行時のインスタンスのID",  " NODE\_INSTANCE\_NO": "「Conductor pause」のNodeインスタンスID",  "RESULTCODE": "結果コード",  "RESULTINFO": "詳細情報"  }  } |
|
|

## Conductor作業確認を対象としたRestAPI

Conductorの操作をRestAPIで行うことができます。

利用可能な機能は、メニューグループ「Conductor」の、「Conductor作業実行」メニュー、「Conductor作業確認」メニューに相当する操作です。

表 5‑22　対象メニュー一覧

|  |  |  |
| --- | --- | --- |
| メニューグループ | メニュー名 | メニューID |
| Conductor | Conductor作業確認 | 2100180005 |

### リクエストの形式

下記の情報でHTTPリクエストを発行します。

・パス

https://<HostName>:<Port>/default/menu/07\_rest\_api\_ver1.php?no=メニューID

メニューIDは**「**表 4‑19　X-Commandに指定可能なパラメータ一覧**表 5‑24　X-Commandに指定可能なパラメータ一覧」**を参照してください**。**

・HTTPヘッダ

表 5‑23 HTTPヘッダパラメータ一覧

|  |  |
| --- | --- |
| HTTPヘッダ | 説明 |
| Content-Type | “application/json”を指定する。 |
| Authorization | ITAの認証要メニューにアクセスする場合は、  「ログインID」と「パスワード」\* を、半角コロン(:)で結合して、base64encodeをした値、を指定。 |
| X-Command | INFO  の1つが選択可能 |

X-Commandに指定可能なパラメータ

表 5‑24　X-Commandに指定可能なパラメータ一覧

|  |  |  |  |
| --- | --- | --- | --- |
| X-Command | 説明 | 対象画面 | メニューID |
| INFO | Conductorの状態確認をし、ステータスを返します。 | Conductor作業確認 | 2100180005 |

以下では、それぞれのX-comanndパラメータについての説明を行います。

### レスポンスの項目

以下では、それぞれのX-comannd実行時のレスポンス項目についての説明を行います。

表 5‑25　レスポンス項目一覧

|  |  |
| --- | --- |
| 項目名 | 備考 |
| CONDUCTOR\_INSTANCE\_ID | CONDUCTORインスタンスに対する操作に使用 |
| RESULTCODE | コマンド実行の成否のコード  000：正常終了 |
| RESULTINFO | 詳細情報 |

### INFO

Conductor実行時のインスタンスIDを指定して、実行時の情報を取得します。

・パラメータ

　　以下をJSON形式でcontentに指定してください。

表 5‑26　Conductor実行情報取得パラメータ表

|  |  |
| --- | --- |
| パラメータ名 | 設定値 |
| CONDUCTOR\_INSTANCE\_ID | ConductorインスタンスID※ |

※EXECUTEの返り値で取得したものです。

・レスポンス

　　返されるレスポンスには、JSON形式で格納されています。

|  |
| --- |
| {  "status": "SUCCEED",  "resultdata": {  "CONDUCTOR\_INSTANCE\_INFO": {  "CONDUCTOR\_INSTANCE\_ID": 1,  ・・・（中略）項目は以下①参照・・・  },  "NODE\_INFO": [  "node-1": {  "NODE\_NAME": "11",  ・・・（中略）項目は以下②参照・・・  }  }  　　　　　　・・・・  ],  "RESULTCODE": "000",  "RESULTINFO": ""  }  } |
|
|

1. CONDUCTOR\_INSTANCE\_INFOに格納されるConductorインスタンスの情報配列

表 5‑27　インスタンス配列表

|  |  |
| --- | --- |
| キー | 内容 |
| CONDUCTOR\_INSTANCE\_ID | ConductorインスタンスID |
| CONDUCTOR\_CLASS\_NO | このインスタンスの元クラスのID |
| STATUS\_ID | 実行時ステータス　詳細は以下　表 5‑ |
| EXECUTION\_USER | 実行ユーザー |
| ABORT\_EXECUTE\_FLAG | 緊急停止発令フラグ　未発令：１　発令済み：２ |
| OPERATION\_NO\_IDBH | 登録オペレーションID |
| OPERATION\_NAME | 登録オペレーション名 |
| TIME\_BOOK | 予約日時 |
| TIME\_START | 開始日時 |
| TIME\_END | 終了日時 |

1. NODE\_INFOに格納されるNodeインスタンス情報配列

表 5‑28　Nodeインスタンス情報表

|  |  |
| --- | --- |
| キー | 内容 |
| NODE\_NAME | Node名 |
| NODE\_INSTANCE\_NO | NodeインスタンスNo |
| NODE\_TYPE\_ID | NodeタイプID 詳細は以下　表 5‑3 |
| STATUS | Nodeのステータス 詳細は以下　表 5‑ |
| SKIP | スキップが設定されている　：2 |
| TIME\_START | 開始日時 |
| TIME\_END | 終了日時 |
| OPERATION\_ID | 個別に指定したオペレーションID |
| OPERATION\_NAME | 個別に指定したオペレーション名 |

1. ID対応表

表 5‑29　Conductorインスタンスの実行時ステータスID対応表

|  |  |
| --- | --- |
| ID | ステータス |
| 1 | 未実行 |
| 2 | 未実行(予約) |
| 3 | 実行中 |
| 4 | 実行中(遅延) |
| 5 | 正常終了 |
| 6 | 緊急停止 |
| 7 | 異常終了 |
| 8 | 想定外エラー |
| 9 | 予約取消 |

表 5‑30　オーケストレータのID対応表

|  |  |
| --- | --- |
| ID | ステータス |
| 3 | Ansible Legacy |
| 4 | Ansible Pioneer |
| 5 | Ansible Legacy Role |
| 10 | Terraform |

表 5‑31　NodeタイプID対応表

|  |  |
| --- | --- |
| ID | ステータス |
| 1 | Conductor start |
| 2 | Conductor end |
| 3 | Movement |
| 4 | Conductor call |
| 5 | Parallel branch |
| 6 | Conditional branch |
| 7 | Parallel merge |
| 8 | Conductor pause |
| 10 | Symphony call |

表 5‑32 Nodeインスタンスの実行時ステータスID対応表

|  |  |
| --- | --- |
| ID | ステータス |
| 1 | 未実行 |
| 2 | 準備中 |
| 3 | 実行中 |
| 4 | 実行中(遅延) |
| 5 | 実行完了 |
| 6 | 異常終了 |
| 7 | 緊急停止 |
| 8 | 保留中 |
| 9 | 正常終了 |
| 10 | 準備エラー |
| 11 | 想定外エラー |
| 12 | Skip完了 |
| 13 | Skip後保留中 |
| 14 | Skip終了 |

# Movement利用編

## Movement作業実行を対象としたRestAPI

Movementの操作をRestAPIで行うことができます。

利用可能な機能は、以下メニューグループに該当する「作業実行」、「作業状態確認」に相当する操作です。

表 6‑1 作業実行、状態確認メニュー一覧

|  |  |  |
| --- | --- | --- |
| メニューグループ | メニュー名 | メニューID |
| Ansible-Legacy | 作業実行 | 2100020111 |
| 作業状態確認 | 2100020112 |
| Ansible-Pioneer | 作業実行 | 2100020211 |
| 作業状態確認 | 2100020212 |
| Ansible-LegacyRole | 作業実行 | 2100020312 |
| 作業状態確認 | 2100020313 |
| Terraform | 作業実行 | 2100080009 |
| 作業状態確認 | 2100080010 |

### リクエストの形式

下記の情報でHTTPリクエストを発行します。

・パス

https://<HostName>:<Port>/default/menu/07\_rest\_api\_ver1.php?no=メニューID

メニューIDは**「表6-3　X-Commandに指定可能なパラメータ一覧」**を参照してください**。**

・HTTPヘッダ

表 6‑2 HTTPヘッダパラメータ一覧

|  |  |
| --- | --- |
| HTTPヘッダ | 説明 |
| Content-Type | “application/json”を指定する。 |
| Authorization | ITAの認証要メニューにアクセスする場合は、  「ログインID」と「パスワード」\* を、半角コロン(:)で結合して、base64encodeをした値、を指定。 |
| X-Command | EXECUTE  CANCEL  SCRAM  の3つが選択可能 |

X-Commandに指定可能なパラメータ

表 6‑3　X-Commandに指定可能なパラメータ一覧

|  |  |  |  |
| --- | --- | --- | --- |
| X-Command | 説明 | 対象画面 | メニューID |
| EXECUTE | 予約/作業実行を行います。 | 作業実行 | 2100020111  2100020211  2100020312  2100080009 |
| CANCEL | 予約取り消しを行います。 | 作業状態確認 | 2100020112  2100020212  2100020313  2100080010 |
| SCRAM | 緊急停止を行います。 | 作業状態確認 |

以下では、それぞれのX-comanndパラメータについての説明を行います。

### レスポンスの項目

以下では、それぞれのX-comannd実行時のレスポンス項目についての説明を行います。

表 6‑4　レスポンス項目一覧

|  |  |
| --- | --- |
| 項目名 | 備考 |
| EXECUTION\_NO | 作業Noに対する操作に使用 |
| RESULTCODE | コマンド実行の成否のコード  000：正常終了  001：実行不可  002：予約取消不可  003：緊急停止不可 |
| RESULTINFO | 詳細情報 |

### EXECUTE

Movementクラスとオペレーションを指定して、作業実行を行います。予約日時の指定や、実行モード（ドライラン/実行）を指定できます。

ができます。

・パラメータ

　以下をJSON形式でcontentに指定してください。

表 6‑5 Movement実行パラメータ一覧

|  |  |
| --- | --- |
| パラメータ名 | 設定値 |
| MOVEMENT\_CLASS\_ID | MovementクラスID |
| OPERATION\_ID | オペレーションID |
| PRESERVE\_DATETIME | 予約日時（YYYY/MM/DD ｔｔ:ｍｍ） |
| RUN\_MODE | 1：実行、  2：ドライラン |

|  |
| --- |
| **例）JSON記述例** |
| {  "MOVEMENT\_CLASS\_ID": 1,  "OPERATION\_ID": 1,  "PRESERVE\_DATETIME": "2019/12/24 15:44",  "RUN\_MODE": 1  } |
|
|

・レスポンス

　　返されるレスポンスには、JSON形式で格納されています。

|  |
| --- |
| {  "status": "SUCCEED",  "resultdata": {  "EXECUTION\_NO": "作業No",  "RESULTCODE": "結果コード",  "RESULTINFO": "詳細情報"  }  } |
|
|

### CANCEL

予約日時が登録されている作業Noを指定して、予約実行をキャンセルします。

・パラメータ

　　以下をJSON形式でcontentに指定してください。

表 6‑6 Movement実行パラメータ一覧

|  |  |
| --- | --- |
| パラメータ名 | 設定値 |
| EXECUTION\_NO | 作業No※ |

※EXECUTEの返り値で取得したものです。

・レスポンス

　　返されるレスポンスには、JSON形式で格納されています。

|  |
| --- |
| {  "status": "SUCCEED",  "resultdata": {  "EXECUTION\_NO": "作業No",  "RESULTCODE": "結果コード",  "RESULTINFO": "詳細情報"  }  } |
|
|

### SCRAM

実行されている作業Noを指定して、緊急停止します。

・パラメータ

　　以下をJSON形式でcontentに指定してください。

表 6‑7 Movement実行パラメータ一覧

|  |  |
| --- | --- |
| パラメータ名 | 設定値 |
| EXECUTION\_NO | 作業No※ |

※EXECUTEの返り値で取得したものです。

・レスポンス

　　返されるレスポンスには、JSON形式で格納されています。

|  |
| --- |
| {  "status": "SUCCEED",  "resultdata": {  "EXECUTION\_NO": "作業No",  "RESULTCODE": "結果コード",  "RESULTINFO": "詳細情報"  }  } |
|
|

# 付録

## トラブルシューティング

| **No** | **内容** |
| --- | --- |
| Q-1 | RESTAPIを使用したファイルアップロードに時間かかりすぎる  RESTAPIでの登録処理中にエラーとなる  RESTAPIでファイルアップロード実施中に、WEB画面上の表示、動作が重くなる |
| A-1 | PHPのメモリ設定が足りないことが考えられます。  PHP設定ファイル「php.ini」内の次のパラメータの値を見直して、割り当て可能な最大値を設定してください。  ・memory\_limit PHPに割り当て可能なメモリ  ・post\_max\_size postデータに許可される最大サイズ |